

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平9-50536

(43)公開日 平成9年(1997)2月18日

(51)Int.Cl.*	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 T	13/00		G 0 6 F	15/62
	11/80			3 4 0 A
	17/40			3 2 0 M
				3 5 0 K

審査請求 有 請求項の数4 O L (全 9 頁)

(21)出願番号 特願平8-134755

(22)出願日 平成8年(1996)5月29日

(31)優先権主張番号 特願平7-133287

(32)優先日 平7(1996)5月31日

(33)優先権主張国 日本 (J P)

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 中川 滋雄

東京都港区芝五丁目7番1号 日本電気株

式会社内

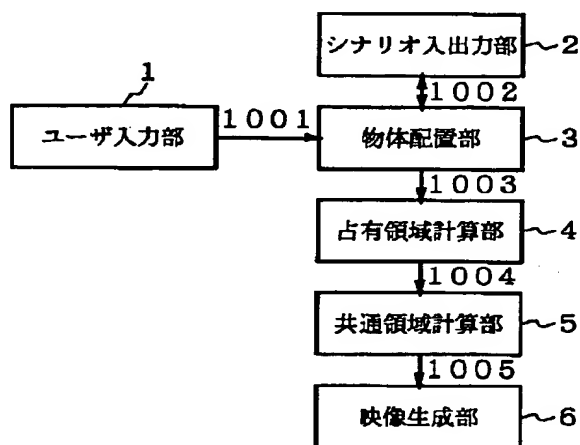
(74)代理人 弁理士 山川 政樹

(54)【発明の名称】 映像表示方法および映像表示装置

(57)【要約】

【課題】 シナリオデータの作成において、登場物体の衝突や干渉の回避が容易にできるようにすることを目的とする。

【解決手段】 物体配置部3により登場物体の配置を定めた後、占有領域計算部4により、動作の開始から終了までの間に、物体が通過する領域を占有領域として求める。そして、共通領域計算部5により、全ての物体について求めた占有領域の間の共通領域を計算する。



1

## 【特許請求の範囲】

【請求項1】 複数の物体を空間中に配置して、フレームごとに前記物体の画像を生成することで一連の映像を表示する映像表示方法において、

第1、第2の物体の幾何属性と、前記第1、第2の物体の変化を含む前記第1、第2の物体それぞれの動きと、前記それぞれの動きを開始する第1、第2の位置および開始する第1、第2の時刻とを第1、2のシナリオとしてそれぞれ定義し、

前記第1、2の位置および第1、2の時刻において、前記第1、2のシナリオを用いて前記第1、2の物体が前記動きを行う際に占有する領域の系列の論理和を近似して前記第1、2の物体が前記第1、2のシナリオに基づいて移動する際の第1、2の占有領域をそれぞれ計算し、

前記第1の占有領域と前記第2の占有領域とが交わる共通領域を計算し、

前記第1、2の占有領域および共通領域とを映像として表示することを特徴とする映像表示方法。

【請求項2】 請求項1記載の映像提示方法において、前記第1、2のシナリオのうち、登場物体の幾何属性と動作データとを変更し、

変更した結果を用いて前記第1の占有領域と前記第2の占有領域とが交わる共通領域を計算し、

前記第1、2の占有領域および共通領域を映像として表示することを特徴とする映像表示方法。

【請求項3】 複数の物体を空間中に配置して、フレームごとに前記物体の画像を生成することで一連の映像を表示する映像表示装置において、

ユーザが入力する指示を取り込むユーザ入力部と、

前記物体の幾何属性、前記物体の変化を含む前記物体の動き、前記動きを開始する位置および開始する時刻からなるシナリオを入力するシナリオ入出力部と、

前記ユーザが入力した指示により前記シナリオを変更して出力する物体配置部と、

前記物体配置部が出力したシナリオにより、前記物体が前記位置および時刻において前記動きを行う際に占有する領域の系列の論理和を近似して前記物体が前記シナリオに基づいて移動する際の占有領域を計算する占有領域計算部と、

前記占有領域および他の物体についての占有領域ととの間の共通領域を計算する共通領域計算部と、

前記占有領域および前記共通領域を表示する映像生成部とを備えたことを特徴とする映像表示装置。

【請求項4】 請求項3記載の映像提示装置において、前記物体配置部の内部に格納されているシナリオのうち、登場物体の幾何属性を、前記ユーザ入力部からの指示に基づいて変更して更新する幾何属性変更部と、前記物体配置部の内部に格納されているシナリオのうち、登場物体の動作データを、前記ユーザ入力部からの

2

指示に基づいて変更して更新する動作変更部とを備えたことを特徴とする映像表示装置。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、ユーザの操作に基づいて変化するアニメーション映像を生成して表示する映像表示方法および映像表示装置に関する。

【0002】

【従来の技術】3次元コンピュータグラフィックス(CG)映像を実時間で生成することにより、対話型の3次元CGアニメーションを実現することができる。これは、ユーザの入力に応じて、アニメーション中の登場物体の動作または動きを変えるものである。対話型の3次元CGアニメーションの利用例としては、以下のものが挙げられる。

①フライト/ドライブシミュレーション

②3次元CGゲーム

③建築物内部を視点移動するウォークスルーシミュレーション

④機械、人体、分子構造などの3次元物体のビジュアル・プレゼンテーションなどのアプリケーション

【0003】このような、対話型のアニメーションにおいては、動作の実行手順を物体ごとに定める方法(スクリプトアニメーション)がある。この方法は、登場物体の動作の実行手順や実行のタイミングを記述するものである。これは、スクリプトと呼ぶアニメーション専用のプログラミング言語によって定められたものである。スクリプトアニメーションでは、まず、動作の種類や実行の順序、そして動作実行の開始条件を物体ごとに記述する。また、アニメーション実行時にユーザの入力などに応じて各スクリプトを実行し、各登場物体の動作の種類や登場物体の位置および動作の実行タイミングを決定している。そして、決定した登場物体の動作に基づいて、アニメーション映像を生成する。

【0004】このスクリプトアニメーションでは、登場物体の動作を登場物体にローカルな座標系および時間軸で記述しておく。そして、そのローカルな座標形および時間軸をグローバルな座標系に対してマッピングし、最終的な対話型アニメーションを生成する。このため、スクリプトアニメーションでは、登場物体の動作位置や動作手順をグローバルな座標系や時間軸に対してあらかじめ定めずに、ユーザの操作によって実行時にそれらを決定することができる。この結果、スクリプトアニメーションでは、対話的なアニメーション生成処理を実現できる。

【0005】スクリプトアニメーションの詳細については文献1(1982年、クレイグ・ダブリュー・レイノルズ、「コンピューター・アニメーション・ウィズ・スクリプト・アンド・アクターズ」、コンピューター・グラフィックス、第16巻 第3号、エイシーエムシググラフ、

289頁-296頁、[Craig W. Reynolds, Computer Animation with Scripts and Actors, Computer Graphics, Vol. 16, No. 3, July 1982, ACM SIGGRAPH, pp. 289-296] )に記載がある。

#### 【0006】

【発明が解決しようとする課題】従来は以上のように構成されていたので、以下に示すような問題があった。前述したスクリプトアニメーションに見られるような対話型アニメーションでは、以下のものをシナリオデータとして記述している。すなわち、登場物体の動作データ、動作手順、登場物体の空間中の配置、および、動作開始のタイミングである。これらのデータの作成の際に、各登場物体の動作データ、空間中の配置、または、動作開始のタイミングの設定によっては、できあがったアニメーションを実行させると、登場物体同士の衝突または干渉が発生してしまう場合がある。

【0007】このような登場物体間の衝突を回避するために、従来では、以下に示す方法が採られていた。まず、あらかじめ登場物体を互いに衝突しない距離を保つように配置しておく。またあるいは、登場物体の動作データを、互いに衝突しないように作成しておくなどの方法がある。しかし、これらの方法ではユーザがデータを修正したときに、衝突が回避されるかどうかを知ることができない。衝突回避を確認するためには、実際にアニメーションを実行して、生成された映像を観察してテストする必要がある。そして、この方法では衝突を回避したデータを作成するまでにこのテストを何度も繰り返す必要がある。

【0008】すなわち、実際にアニメーションを実行する場合、ある実行下では、全ての登場物体が全てそのシナリオで規定されている動作をするわけではない。ある登場物体は、あるアニメーション実行状況では全く動作しない場合もある。このため、実際にアニメーションを実行して衝突状態を検出する場合、あらゆる場合を想定し、複数の状況でアニメーションを実行し、生成された映像を観察して衝突状態を検出することとなる。以上のことにより、従来では、シナリオデータの作成には、多大な工数が必要となるという問題があった。

【0009】この発明は、以上のような問題点を解消するためになされたものであり、シナリオデータの作成において、登場物体の衝突や干渉の回避が容易にできるようにすることを目的とする。

#### 【0010】

【課題を解決するための手段】この発明の映像表示方法は、まず、第1、第2の物体の幾何属性と、第1、第2の物体の変化を含む第1、第2の物体それぞれの動きと、それぞれの動きを開始する第1、第2の位置および開始する第1、第2の時刻とを第1、2のシナリオとしてそれぞれ定義する。次に、それら第1、2の位置および第1、2の時刻において、第1、2のシナリオを用い

て第1、2の物体が動きを行う際に占有する領域の系列の論理和を近似し、第1、2の物体が第1、2のシナリオに基づいて移動する際の第1、2の占有領域をそれぞれ計算する。そして、得られた第1の占有領域と第2の占有領域とが交わる共通領域を計算し、第1、2の占有領域および共通領域とを映像として表示するようにした。すなわち、登場する物体の配置を定めた後、動作の開始から終了までの間に、物体が通過する領域を占有領域として求め、全ての物体について求めた占有領域の間の共通領域を計算するようにした。このため、ユーザは、共通領域が生じれば、物体間で衝突が発生することを予め知ることができる。そして、共通領域が無くなるように物体の再配置を行えば、衝突を回避したシナリオを作成できる。

【0011】また、この発明の映像表示装置は、ユーザが入力する指示を取り込むユーザ入力部と、物体の幾何属性、物体の変化を含む物体の動き、動きを開始する位置および開始する時刻からなるシナリオを入力するシナリオ入力部と、ユーザが入力した指示によりシナリオを変更して出力する物体配置部と、物体配置部が出力したシナリオにより、物体が位置および時刻において動きを行う際に占有する領域の系列の論理和を近似して物体がシナリオに基づいて移動する際の占有領域を計算する占有領域計算部と、占有領域および他の物体についての占有領域との間の共通領域を計算する共通領域計算部と、共通領域とその共通領域を共有する物体およびその物体の占有領域を表示する映像生成部とを備えるようにした。すなわち、登場する物体の配置を定めた後、動作の開始から終了までの間に、物体が通過する領域が占有領域として求められ、全ての物体について求めた占有領域の間の共通領域が計算される。このため、ユーザは、共通領域が生じれば物体間で衝突が発生することを予め知ることができる。そして、共通領域が無くなるように物体の再配置を行えば、衝突を回避したシナリオを作成できる。

#### 【0012】

【発明の実施の形態】まず、この発明の実施の形態を説明する前に、この発明の概要について説明する。スクリプトアニメーションでは、まず、以下のものを定義する。

①対話型アニメーションの登場物体に対して、ローカル座標系における並行移動、回転、変形などの時間変化をとともう動き（動作）を定義する。

②登場人物の動作の開始条件とその実行順序とを、動作手順とともに登場物体データとして定義する。

そして、定義した登場物体データを、対話型アニメーションのシナリオデータとして記述する。このとき、登場物体の空間中の位置座標および一連の動作の開始時刻からなる配置データも組み合わせて、シナリオデータとして記述する。

【0013】次に、登場物体の配置データを定めた後、この発明では、その登場物体の動作の開始から終了までの間に登場物体が通過する領域を、占有領域として計算して求める。そして、対話型アニメーションに登場する登場物体について、それぞれ前述の占有領域を計算し、各占有領域の間に共通の領域が生じるかどうかを調べる。この調査の中で、共通領域が生じる場合には、これにかかわる占有領域を持つ登場物体の間での衝突が発生する。このため、この発明では、この共通領域と共通領域にかかわる登場物体、および共通領域を共有する占有領域のいずれかまたは全部の情報を、映像としてユーザに示すようにした。

【0014】その結果、ユーザは、共通領域の発生を映像として認知することによって、実際にアニメーションを実行しなくても、登場物体間の衝突を知ることができる。そして、ユーザは、登場物体間の衝突を認知した後、生じた共通領域が消滅するまで、登場物体の空間座標または動作タイミングなどのデータを連続して変化させれば、登場物体同士の衝突がない状態を得ることができる。また、この修正操作の間、ユーザは自らの操作が反映した占有領域および共通領域の変化の様子を、リアルタイム表示映像として観察することができる。このため、登場物体間の衝突を回避する共通領域の消滅のためには、どのような修正操作が有効かを、容易に知ることができる。

【0015】以上説明したように、この発明においては、ユーザは、実際にアニメーションを実行することなく、登場物体間の衝突を検出することができる。すなわち、ユーザが、共通領域が消滅するまで登場物体のデータ修正作業を行うことで、登場物体間での衝突を回避しながら、新規に登場物体を空間中に配置することができる。さらに、何度も実際にアニメーションの実行テストを繰り返すことなく、衝突回避のための登場物体の配置データの修正を行うことができる。この結果、ユーザは大幅に少ない工数で、対話型アニメーションのシナリオデータの作成を行うことができる。

【0016】以下この発明の実施の形態を図を参照して説明する。

実施の形態1. 図1は、この発明の第1の実施の形態における映像表示装置の構成を示す構成図である。以下では、空間および登場物体を3次元で定義し、3次元コンピュータグラフィックス技術を用いてアニメーション映像を生成する場合を例として説明する。しかし、本発明は、例えば、スプライト表示方式のTVゲームのように、2次元の平面でアニメーションを生成する場合にも全く同様に実施できる。一般に、 $N$  ( $N=1, 2, 3, \dots$ ) 次元の空間でアニメーション映像を生成する場合に、この発明を適用して実施することができる。

【0017】図1に示されるように、ユーザ入力部1は、ユーザが操作するマウス、キーボード、3次元座標

入力装置などが出力するデータを取り込み、ユーザ入力1001として出力する。シナリオ入出力部2は、以下で説明するシナリオデータを外部から取り込んで、シナリオ1002として物体配置部3に出力する。また、このシナリオ入出力部2は、物体配置部3が出力するシナリオ1002を外部へ出力する。

【0018】なお、ユーザ入力部1およびシナリオ入出力部2において取り込むデータは、例えば、以下に示すものを用いることができる。

- ①ユーザがキーボード、マウス、3次元座標入力装置などを操作して得られたデータ
- ②磁気テープ、磁気ディスク等の記憶装置にあらかじめ記録しておいたデータ
- ③同一の計算機上で稼働する別のプログラムにより生成してメモリ上に格納したデータ
- ④通信回線によって接続した他の計算機または映像表示装置との通信結果をメモリ上に格納したデータ

【0019】ここで、まず、登場物体が通過する領域である占有領域の表示に関する説明の前に、シナリオ1002に関して説明する。シナリオ1002は、以下に説明するように、対話型アニメーションに登場する全ての登場物体について、(1)幾何属性、(2)動作データ、(3)動作手順、および(4)配置データの組をそれぞれ記述したデータである。なお以下の説明では、3次元グラフィックスを使用した映像生成において必要な光源および視点は、それぞれ配光特性および視野特性などの特殊な幾何属性を持つ登場物体とし、他の登場物体と同様に扱うものとする。

【0020】(1)幾何属性

幾何属性には、まず、登場物体の形状や形状表面の輝度や材質がある。また、テクスチャマッピングするテクスチャ画像、テクスチャ画像と形状表面との対応関係などがある。このなかで、形状データは、たとえば、物体座標系におけるポリゴンの座標データ、またはスプライン曲面の制御点データなどにより記述する。また、形状表面の材質は色輝度値や反射係数などにより記述する。また、関節を持つ物体など、登場物体がさらに部分的な登場物体の階層的な接続により構成される場合には、階層関係の管理情報と、部分登場物体間の位置関係も幾何属性として記述する。

【0021】(2)動作データ

動作データとは、登場物体の空間中における動きの情報を、登場物体の属する座標系から、その上位の座標系またはワールド座標系への対応関係の時間変化によって定義したものである。一例として、登場物体の属する物体座標系から上位座標系への平行移動、回転、スケーリング(拡大、縮小)、および、その組み合わせを行う同次座標行列 $M$ について、フレーム間隔ごとの時刻( $t_1, t_2, \dots, t_{n-1}, t_n$ )のサンプリング記録( $M(t_1), M(t_2), \dots, M(t_{n-1}), M(t$

7

n))を、動作データとして定義することができる。フレーム間隔として、例えばTV放送(NTSC方式)のフィールド時間(1/60秒)またはフレーム時間(1/30秒)などを設定することで、滑らかな動きの映像を生成できる。

【0022】また、時刻 $t$ をパラメータとする関数 $D$ によって、時刻 $t$ での同次座標行列に等価な対応関係 $D(t)$ を定義し、関数 $D$ およびパラメータ $t$ の組み合わせを物体の動作データとしてもよい。たとえば、スプライン曲線関数( $D$ )とその入力パラメータ( $t$ )とによ

って、物体の動きを記述してもよい。  
 【0023】この、スプライン曲線関数 $D(t)$ は、実数パラメータ $t$ により空間中の1点の座標値( $x(t)$ ,  $y(t)$ ,  $z(t)$ )を定める関数である。このスプライン曲線関数の一例として、CAD/CAMの分野において一般にNURBS曲線として知られる関数を使用できる。NURBS曲線については、文献2(J. D. Foley and A. van Dam and S. K. Feiner and J. F. Hughes, Computer Graphics: Principles and Practice (Second Edition), 1990, Addison Wesley)「1990年、ジェイ・ディー・フォーリー他”コンピュータ・グラフィックス・プリンシプルズ・アンド・プラクティス(第2版)”、アディソン・ウエズレイ」の501~504頁に記載がある。

【0024】さらに、部分物体間の接続に対して対応する動き情報を定義し、複数の動き情報の集合によって登場物体の動作を定義することができる。これは、登場物体が、例えば動物のように部分的な物体が関節によって接合している場合や、あるいは登場物体をさらに複数の部分により階層的に構成する場合である。特に、関節によって部分物体を結合して構成する物体の場合は、部分物体を定義した物体座標系間の関節における回転角の時間変化によって、動作を記述してもよい。

【0025】前述した登場物体の物体座標系内における移動の他に、登場物体の形状の変形も動作として扱う。時間変化を伴う形状変形を表現するために、例えば形状変形させるための変形アルゴリズムとパラメータの変化を記述することができる。一例を挙げれば、登場物体に対してアフィン変換などの一次変換を行って変形させる場合には、時系列順に記述したアフィン変換行列の集合によって形状変形動作を記述することができる。

【0026】ここで、3次元のアフィン変換操作については、前述した文献2の213~226頁に記載されている。以下、アフィン変換による形状変化を簡単に説明する。まず、時刻 $t = (t_1, t_2, \dots, t_{n-1}, t_n)$ における3次元のアフィン変換行列を $T = (T(t_1), T(t_2), \dots, T(t_{n-1}), T(t_n))$ とする。このとき、各時刻 $t$ における登場物体の形状は、アフィン変換によって、形状を構成するポリゴンまたはスプラインパッチを記述する座標値 $p$ に対して、そ

8

れぞれ時刻 $t$ における $T(t)$ を演算した座標値 $p' = T(t)p$ となる。

【0027】以上説明した登場物体の動き情報(動作データ)は、動きの計算アルゴリズムが固定であれば入力となるパラメータだけで記述することができる。例えば、前述のNURBS曲線関数を固定した計算アルゴリズムとする場合では、NURBS曲線の制御点を記述するパラメータだけで動き情報を与えることができる。また、計算アルゴリズムを変更する場合には、アルゴリズムまたはその種別とパラメータとを組み合わせで動作データとする。

【0028】なお、登場物体間でメッセージ通信を行ってアニメーションを進行させるような対話型アニメーションでは、メッセージの送信を動作の一種として定義してもよい。登場物体間でメッセージ通信を行うイベント駆動型のスクリプトアニメーションの例については、文献3(特開平3-168877号公報)に記載がある。

【0029】(3)動作手順

対話型アニメーションでは、登場物体に複数の動作をあらかじめ定義しておき、同時にそれら動作の実行の手順を動作手順として記述する。動作手順の記述に際しては、前述のスクリプトアニメーションではプログラミング言語を用いている。このため、登場物体に内部状態を与えるために、状態遷移のための内部変数を設けている。

【0030】なお、動作手順の記述は、具体的には、C言語やLISP言語、あるいはこれらに準じた手続き型のプログラミング言語により行なうことができる。対話型アニメーションの作成者はこれらの言語により登場物体の動作データ、動作アルゴリズムの種別、動作の実行タイミング、動作の実行順序および開始条件を記述できる。

【0031】ここで、登場物体の動作手順について、一般的なスクリプトアニメーションなどの対話型アニメーションで用いられる方法として説明する。一般に対話型アニメーションでは、動作手順は、複数の動作中から実行すべき動作を決定する関数の集合である。なお、複数の動作は、外界からのイベントまたはメッセージを入力とするものである。例えば、前述の動作手順の記述においては、動作の開始条件が動作を決定すべき関数に相当する。逐次実行、ループ、条件分岐などの制御構造に従って開始条件を記述することで、対話型アニメーションの動作手順を定めることができる。そして、この動作手順は、動作の制御手順(例:逐次実行、条件分岐、繰り返し)およびそれら動作の開始条件を、プログラムで記述したデータとして定義されている。

【0032】(4)配置データ

登場物体の幾何属性、動作データは、それぞれ物体に対してローカルな座標系で定義する。そして、そのローカルな座標系を、対話型アニメーションの実行時にワール

ド座標系などのグローバルな座標系に配置する。さらに、グローバルな時間軸における登場物体の動作開始時刻を定める。そして、その配置に基づき最終的なアニメーション映像を生成する。このため、登場物体には、グローバルな座標系および時間軸に対して配置する際の座標値を、配置データとして定義する。

【0033】一例として、登場物体の配置データは、以下の座標や時刻などの組によって記述することができる。

①ワールド座標系の原点に対する空間的な並行移動量と座標値

②座標軸まわりの回転を記述したアフィン変換行列

③グローバルな時間軸における登場物体の一連の動作における開始時刻

なお、以下に示すような場合には、グローバルな座標系ではなく、階層関係の直上に相当する座標系に対するデータとして配置データを記述し、登場物体の間の配置関係を設定することができる。例えば、移動する台の上に置かれた登場物体の場合のように、独立した登場物体がさらに他の登場物体に対して階層的な関係に配置される場合である。

【0034】シナリオ1002は、以上に示したように構成され、物体配置部3内部に格納されている。そして、物体配置部3は、ユーザ入力1001に基づいて、記述されている登場物体の配置データを変更し、内部に格納したシナリオを更新する。物体の配置に際しては、まずその更新したシナリオ中から配置データを変更する登場物体を選択する。物体の選択に際しては、たとえば対話型アニメーションにおける任意の時刻の映像を生成し、ユーザ入力1001のマウスの座標値から、ピッキングなどの計算を行うなどの方法で、ユーザの選択する登場物体を知ることができる。

【0035】ここで、ピッキングは、ユーザのポインティングデバイスの操作結果に基づいて操作の対象となった物体を検索して参照するための計算手法である。たとえば、CRTの画面上に表示された物体をユーザがマウスでクリックした際に、クリックの対象となった物体を知るために行なう。ここで、CRTは対話型アニメーションが表示されているものである。また、マウスはポインティングデバイスであり、CRT中に表示されているものをクリックすることで選択指示ができるものである。なお、ピッキングの実現手法に関しては、前述した文献2の48～50頁に記載がある。

【0036】登場物体の配置データのうち、物体の空間的な座標値の変更は、マウスによる物体のドラッグ操作などの方法による、ユーザ入力1001で行うことができる。また、登場物体のグローバルな時間軸における動作の系列の開始時刻は、ユーザ入力1001より直接時刻の数値を入力した値を用いても良い。また、対話型アニメーションを実行して映像を生成し、登場物体を

配置すべき時刻で実行を停止し、停止した時刻を入力するようにしても良い。

【0037】以上説明したように、物体配置部3は、登場物体の配置データを変更して内部に格納したシナリオを更新した後、更新したシナリオを物体配置情報1003として出力する。この物体配置情報1003は、ユーザの操作による配置データ変更のためのユーザ入力1001が発生する度に、あるいは映像生成のフレームタイムごとに出力するようにしてもよい。

【0038】また、物体配置部3は、ユーザによる一連の登場物体間の衝突を回避した登場物体配置が完了した場合、外部に対してシナリオ1002として物体配置情報を出力することもできる。これは、物体配置部3が、物体配置情報をシナリオ入出力部2に出力することによってなされる。あるいは、物体配置部3は、その作業を行う途中の任意の時点で、外部に対してシナリオ1002として物体配置情報を出力することもできる。

【0039】以下、登場物体が通過する領域である占有領域の表示について説明する。まず、前述したことにより、物体配置情報1003が与えられると、占有領域計算部4は、これに基づき、登場物体の幾何形状および動作データによって得られる占有領域を計算する。ここで、空間占有領域の計算について説明する。図2は、登場物体の幾何属性、動作データ、および配置データによって得られる、登場物体の占有領域の状態を示す斜視図である。

【0040】図2において、グローバルな座標系XYZの時刻 $t = T_0$ において、登場物体31は動作データ35が定義されており、その結果時刻 $t = T_1, T_2$ においてそれぞれ位置32、33に移動する。このとき、登場物体31は、動作データ35を実行することにより、グローバルな座標系における占有領域34の内部を動くことになる。占有領域とはすなわち、次に示すことによって得られた空間領域とすることができる。まず、配置データに基づいて、登場物体をその動きデータごとグローバルな座標系に変換する。その後、登場物体を動きデータに基づいてスイープ（掃引）操作を行う。

【0041】スイープ操作に関しては前述した文献2の540～542頁に記載されている。この例では、2次元図形を3次元空間中で移動させ、移動の各時点における図形を包絡して3次元形状を計算している。本発明の空間占有領域の計算においては、スイープ操作により、登場物体の3次元形状を3次元空間内において動きデータに基づいて移動する。そして、その移動における移動の各時点における形状を包絡し、空間占有領域を計算する。

【0042】図2で説明すると、空間占有領域34は、例えば、動作データ35の動作開始時刻 $t = T_0$ から $t = T_2$ までの間の、フレームタイムごとに得られる登場物体31の占める空間領域の系列を、論理和して得られ

る空間領域として計算しても良い。一例をあげると、以下に示す(1)式により、空間占有領域を定義する。なお、式中<OR>は、形状の空間的な論理和演算とする \*

$$V(T0) <OR> V(T0+d) <OR> \dots <OR> V(T2-d) <OR> V(T2) \dots (1)$$

なお、(1)式において、dは映像のフレーム間隔時間(例えば1/30秒)を示している。また、時刻t=(T0, T0+d, ..., T2-d, T2)における登場物体が空間中で占める空間を、V(t)=(V(T0), V(T0+d), ..., V(T2-d), V(T2))とする。

【0044】形状の論理演算には、ソリッドモデリングと呼ばれるCAD/CAM分野の形状定義手法を用いることができる。ソリッドモデリングについては、前述した文献3の535~539頁に記載がある。そして、計算した占有領域は、登場物体の幾何属性と同様に、ポリゴンやスプライン曲面によって近似して表現することができる。なお、占有領域の計算においては、時間的および空間的にサンプリングまたは近似を行っても良い。

【0045】例えば、時刻t=T0, T1, T2でサンプリングした登場物体の形状を、内部に含んで取り囲む最小の直方体(バウンディングボックス)のように、ある程度近似した空間領域として計算してもよい。例えば、前述の(1)式において、フレーム間隔時間dを大きくとって計算することにより、時間的にサンプリングした空間占有領域を計算できる。例えば、dから2dに変更して計算する。この結果、空間占有領域を求めるための計算量を削減することができる。

【0046】この場合、時間的サンプリング間隔はかならずしも一定である必要はない。物体の移動する速度が大きい(位置の時間変化が大きい)ところは、時間サンプリング間隔を細かくしてもよい。また、移動速度が遅い(位置の時間変化が小さい)ところは、時間サンプリング間隔を粗くするように、適応的に設定することもできる。

【0047】また、物体形状の代りに、比較的単純な幾何形状を(1)式のV(T)として使用することで、空間的に近似を行なうことができる。比較的単純な幾何形状とは、物体のバウンディングボックスや物体を取り囲む球形状などである。この結果、空間占有領域を求めるための計算量を削減することができる。なお、空間的な近似と前述の時間的サンプリングは、同時に用いることもできる。

【0048】図2の例では、登場物体31が動作データを1個しかもたない場合について説明している。しかし、登場物体31がユーザとの対話によって複数の動きデータを選択し得る場合には、選択肢となる動きについてそれぞれ占有領域を計算し、計算した領域の論理和を登場物体31の占有領域としてもよい。

【0049】また、登場物体が長い時間または広い空間領域にわたって動作を行うために、占有領域が非常に大

きる。

【0043】

10

きくなる場合がある。このような場合には、例えば、動作データのうち占有領域を計算する時間的な区分を設定する。このようにすれば、占有領域を映像化する際に、ユーザの理解を助けることになる。特に、ある任意の時刻における占有領域を計算する場合には、その時刻以前の動きデータに相当する部分についての占有領域の計算を省略してもよい。

【0050】占有領域計算部4は、以上説明したように、登場物体の占有領域を計算し、計算結果を物体配置情報1003と合わせて空間占有情報1004として出力する。共通領域計算部5は、空間占有情報1004に基づいて、各登場物体の占有領域の間の共通となる領域を計算する。図3は、各登場物体の占有領域およびそれらの間で共通な領域の状態を示す斜視図である。

20

【0051】図3において、占有領域41, 42は、互いにグローバルな座標系XYZにおいて、斜線で示す共通領域43を共有している。共通領域43の計算は、占有領域間の論理積となる空間領域を求めることにより行うことができる。計算した共通領域43は、占有領域の場合と同様に、ポリゴンやスプライン曲面によって近似して表現することができる。この、共通領域の計算は前述したソリッドモデリングにおける論理積演算によって行なうことができる。

30

【0052】共通領域計算部5は、以上説明したように共通領域を計算して、計算結果を空間占有情報1004と合わせて空間共通情報1005として出力する。映像生成部6は、空間共通情報1005に記述された登場物体のシナリオデータ、占有領域、および、共通領域に基づいてフレームごとに画像を生成する。そして、映像生成部6は、一連の生成画像によって、登場物体および上述したその動作による軌跡領域である占有領域を表示する。加えて、映像生成部6は、登場物体同士が衝突する状態を、占有領域が重なっている共通領域として表示する。

40

【0053】映像生成部6における画像生成では、一連の3次元コンピュータグラフィクス技術による画像生成処理を行う。これは、視点や光源を含む登場物体、占有領域、または、共通領域の空間中の位置と形状、および、表面の属性に基づいておこなう。なお、3次元コンピュータグラフィクス技術としては、座標変換、クリッピング、輝度計算、透視投影計算、シェーディング、隠面消去処理などがある。

【0054】このとき、占有領域および共通領域は、それぞれ異なるように表示属性を設定した映像としても良い。例えば、色輝度を変えたり、あるいは半透明にしたり、あるいは領域の輪郭線を表示したりするなどのこと

50



をする。このことにより、登場物体との位置関係およびそれらの領域間の位置関係が、ユーザにとって把握しやすくなる。

【0055】以上示したように、この実施の形態によれば、アニメーション中の登場物体に関連する占有領域と、それら各占有領域の共通領域を表示する。また、ユーザの入力による登場物体の各属性やデータの変更を反映させ、その占有領域や共通領域を変更して表示する。このため、ユーザは、映像生成された占有領域の間に共通領域が発生しないように、登場物体のグローバル空間および時間軸における配置を行うことができる。すなわち、ユーザは、アニメーションを実行することなく、登場物体の衝突を回避した状態の各配置を行うことができる。また、配置を行った後のシナリオを、シナリオ入出力部2を通じて外部へ出力することができる。

【0056】実施の形態2。次に、この発明の第2の実施の形態について説明する。図4は、この発明の第2の実施の形態における映像表示装置の構成を示す構成図である。この実施の形態2における映像表示装置は、上記実施の形態1に対して、幾何属性変更部7と動作データ変更部8とを付加していることを特徴としている。

【0057】幾何属性変更部7は、物体配置部3の内部に格納したシナリオのうち、登場物体の幾何属性を読みだす。また、幾何属性変更部7は、ユーザ入力1001に基づいて、その読みだした幾何属性を変更する。そして、幾何属性変更部7は、変更結果の幾何属性1006によって、物体配置部3の内部に格納したシナリオにおける幾何属性を更新する。

【0058】同様に、動作データ変更部8は、物体配置部3の内部に格納したシナリオの中より登場物体の動作データを読みだす。ついで、動作データ変更部8は、ユーザ入力1001に基づいて読みだした動作データを変更する。そして、動作データ変更部8は、変更結果の動作データ1007によって、物体配置部3の内部に格納したシナリオの動作データを更新する。物体配置部3は、幾何属性変更部7および動作データ変更部8により内部に格納したシナリオが更新されるたびに、登場物体の配置データの変更の場合と同様に物体配置情報1003を出力する。

【0059】以上説明した動作によって、ユーザは、配置データの変更だけではシナリオ中の登場物体間の衝突が回避しきれない場合に、幾何属性および動作データを変更して登場物体の占有領域の形状を変更することができる。この結果、登場物体間の衝突回避のためのシナリオデータの変更作業の自由度を高めることができる。そして、ユーザの意志によりよく従ったシナリオデータの

作成を行うことができるようになる。

【0060】

【発明の効果】以上説明したように、この発明では、まず、第1、第2の物体の幾何属性と、第1、第2の物体の変化を含む第1、第2の物体それぞれの動きと、それぞれの動きを開始する第1、第2の位置および開始する第1、第2の時刻とを第1、2のシナリオとしてそれぞれ定義する。次に、それら第1、2の位置および第1、2の時刻において、第1、2のシナリオを用いて第1、2の物体が動きを行う際に占有する領域の系列の論理和を近似し、第1、2の物体が第1、2のシナリオに基づいて移動する際の第1、2の占有領域をそれぞれ計算する。そして、得られた第1の占有領域と第2の占有領域とが交わる共通領域を計算し、第1、2の占有領域および共通領域とを映像として表示するようにした。

【0061】その結果、本発明によれば、ユーザはそれぞれの登場物体の占有領域間の共通領域の有無によって、登場物体間の衝突をあらかじめ知ることができる。このために、実際にアニメーションを実行することなく登場物体間の衝突を検出することができる。さらに、共通領域が生じる場合には、ユーザは共通領域が消滅するように、登場物体のグローバルな座標系での配置および動作の開始タイミングをリアルタイム生成映像を見ながら変更することができる。このために、登場物体間の衝突回避のために何度も実際にアニメーションの実行テストを繰り返すことがない。この結果、ユーザは、大幅に少ない工数で、対話型アニメーションのシナリオデータの作成を、行うことができるという効果を有する。

【図面の簡単な説明】

【図1】 この発明の1実施の形態における映像表示装置の構成を示す構成図である。

【図2】 登場物体の幾何属性、動作データ、および配置データによって得られる、登場物体の占有領域の状態を示す斜視図である。

【図3】 各登場物体の占有領域およびそれらの間で共通な領域の状態を示す斜視図である。

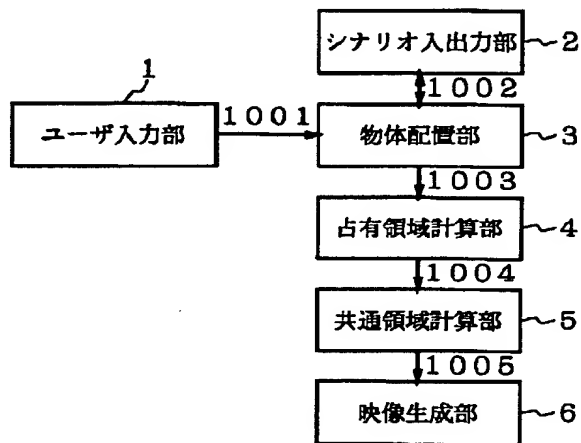
【図4】 この発明の第2の実施の形態における映像表示装置の構成を示す構成図である。

【符号の説明】

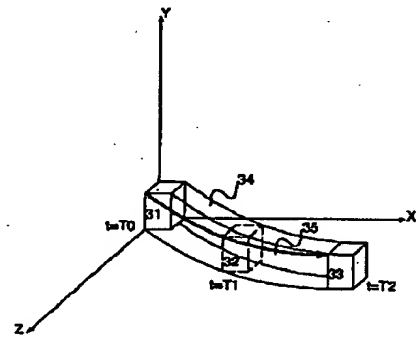
1…ユーザ入力部、2…シナリオ入出力部、3…物体配置部、4…占有領域計算部、5…共通領域計算部、6…映像生成部、7…幾何形状変更部、8…動作データ変更部、1001…ユーザ入力、1002…シナリオ、1003…物体配置情報、1004…空間占有情報、1005…空間共通情報、1006…幾何属性、1007…動作データ。



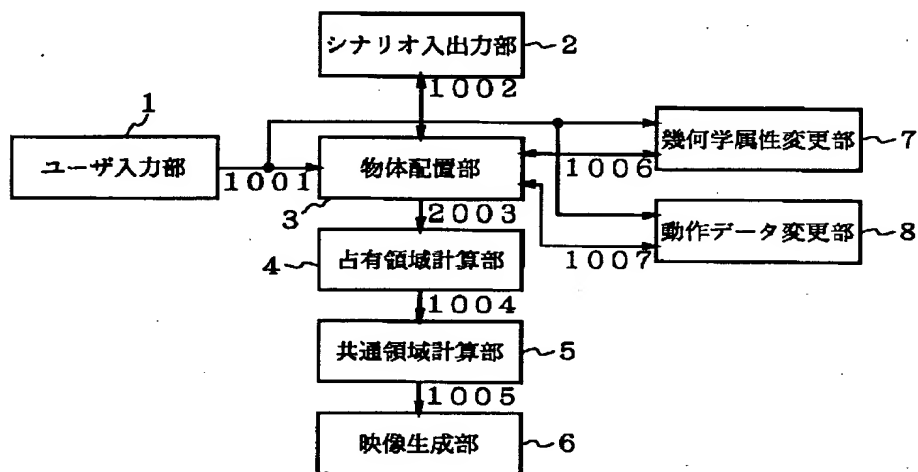
【図1】



【図2】



【図4】



【図3】

